Characterizing Evaluation Practices of Intrusion Detection Methods for Smartphones

Abdullah J. Alzahrani, Natalia Stakhanova, Hugo Gonzalez and Ali A. Ghorbani

Information Security Center of Excellence, Faculty of Computer Science, University of New Brunswick
{*a.alzahrani, natalia, hugo.gonzalez, ghorbani*}*@unb.ca*

Received 28 February 2014; Accepted 15 April 2014;
Publication 2 July 2014

Abstract

The appearance of a new Android platform and its popularity has resulted in a sharp rise in the number of reported vulnerabilities and consequently in the number of mobile threats. Mobile malware, a dominant threat for modern mobile devices, was almost non-existent before the official release of the Android platform in 2008. The rapid development of mobile platform apps and app markets coupled with the open nature of the Android platform triggered an explosive growth of specialized malware and subsequent search for effective defence mechanisms. In spite of considerable research efforts in this area, the majority of the proposed solutions have seen limited success, which has been attributed in the research community to the lack of proper datasets, lack of validation and other deficiencies of the experiments. We feel that many of these shortcomings are due to immaturity of the field and a lack of established and organized practice. To remedy the problem, we investigated the employed experimentation practices adopted by the smartphone security community through a review of 120 studies published during the period between 2008–2013. In this paper, we give an overview of the research in the field of intrusion detection techniques for the Android platform and explore the deficiencies of the existing experimentation practices. Based on our analysis

Journal of Cyber Security, Vol. 3 No. 2, 89–132.
doi: 10.13052/jcsm2245-1439.321

we present a set of guidelines that could help researchers to avoid common pitfalls and improve the quality of their work.

Keywords: intrusion detection, smartphones, mobile malware.

1 Introduction

The rapid evolution of various mobile platforms in the past decade has swiftly brought smartphones to all aspects of our daily life. Such popularity has stimulated underground communities, giving an unprecedented rise to mobile malware. Among the most targeted platforms is the Android platform, mostly due to the ease of use of malicious apps, and the lack of proper defence. According to Kaspersky's estimation, the number of mobile malware targeting the Android platform tripled in 2012, reaching 99% of all mobile malware [37]. Also, they said that in 2013 there are more than 148,427 mobile malware modifications in 777 families and 98.05% of mobile malware found this year targets Android platform [41].

The lack of necessary defence mechanisms for mobile devices has been mostly restricted by the limited understanding of these emerging mobile threats and the resource-constrained environment of smartphones. Indeed, on the one hand, the rapid growth of vulnerabilities for a new and less-studied platform, coupled with the lack of timely access to emergent mobile malware, hinder our abilities to analyze these threats. On the other hand, the resource-constrained environment of smartphones, which is unable to afford computationally intensive operations, presents a significant challenge to the development of intelligent intrusion detection solutions. With mobile phone security quickly becoming an urgent necessity, researchers have started focussing their attention on the problem.

In the past several years the number of studies in the field of mobile phone security has been steadily increasing. In light of recent work around security-related studies in long established domains (e.g. anomaly detection), a lack of scientific rigor has been shown in the experimentation in the majority of these studies [42–43, 47]. As a result, we examine the evaluation practices of a newly appearing field of mobile phone security.

In this paper, we explore research in the area of intrusion detection for the mobile platform published during the period between 2008–2013. Aiming to discover the problems related to experimentation rigor encountered in other fields, we highlight the most common shortcomings and offer a set of guidelines for proper evaluation practice to the smartphone intrusion detection

community. Within this study we give an overview of the common trends in smartphone intrusion detection research highlighting the general focus of the research efforts and the existing gaps. We hope that these efforts will give an insight into the future development of viable intrusion detection mechanisms for mobile devices.

The rest of the paper is organized as follows: in Section 2, we present some related works; in Section 3, we discus intrusion detection in mobile devices; in Section 4, we provide our assessment methodology; in Sections 5–6, we discuss the results of our evaluation; in Section 7, we present a set of guidelines that would help researchers to avoid common pitfalls and improve the quality of their work. Finally, we summarize our conclusion in Section 8.

2 Related Work

With the recent burst of research interest in the area of smartphone security, a number of studies have been aiming to organize and classify the existing efforts. One of the first attempts to summarize the research in the area of security for mobile devices was presented by Enck [24]. A broader study focusing on a variety of mobile technologies (e.g., GSM, Bluetooth), their vulnerabilities, attacks, and the corresponding detection approaches, was conducted by Polla et al. [33]. A more thorough analysis of research in the area of smartphone related to security solutions was offered by Shahzad [46]. Before these major surveys there were several other studies focusing on various aspects of mobile phone security [49, 19, 30, 12].

The lack of clear guidelines for structuring and analyzing existing research in the area of smartphone security has triggered additional efforts aiming to devise a structured taxonomy and provide necessary classification criteria. Among these efforts, there are a taxonomy for classification of smartphone malware detection techniques proposed by Amamra et al. [12], and a classification of attack vectors for smartphones developed by Becher [17].

To complement these research efforts, several study groups have been surveying mobile malware characteristics. Felt et al. [27] evaluated the behavior of 46 mobile malware samples and the effectiveness of existing defence mechanisms. On a broader scope, Zhou et al. [52] gave a detailed characterization of over 1000 Android malware samples.

This paper, on the other hand, steps beyond traditional survey boundaries and takes a critical look at the experimentation practices adopted by the smartphone security community.

3 Specificity of Intrusion Detection in Mobile Devices

Intrusion detection in traditional networks is one of the most well defined and extensively studied fields. Intrusion detection in mobile networks generally falls under an umbrella of this broader domain, and thus the core foundation of intrusion detection generally follows the defined principles. At the same time, there are several specificities that make traditional IDSs not suitable for mobile devices:

- *Constrained resources*: the resource-constrained environment of smartphones puts strict limitations on the usage on the available time, power, and memory resources, essentially dictating what actions the detection system can and cannot afford. As such many of the approaches that require a heavy computational operations (e.g., malware static analysis) are avoided.
- *Mobility*: As opposed to traditional IDSs where an IDS system is permanently stationed on a known network or a host, mobile device IDSs are generally located on a mobile device with some more resource intensive functionality residing on a cloud. Thus as mobile device goes through a variety of networks with often unknown configurations and different security postures, mobile IDS faces various challenges to provide a comprehensive defense for a wide range of threats and conditions.
- *Deployment environment*: One of the security features characterizing modern mobile platforms is the use of sandbox environment that allows to constrain unwanted activity. Since a sandbox is meant to execute untrusted code, trusted native code is generally run directly on a platform. Although sandboxing is generally seen as a desirable intrusion detection technique, it has limitations. Sandboxing is usually less effective in detecting non-generic targeted attacks, e.g., malware designed to be activated on specific user action or to trigger malicious behavior after a period of normal activity.

  Sandboxing is also largely ineffective against another practice, i.e., the use of external code, that have been gaining popularity in mobile apps. This mechanism allows to use legitimate application to load a malicious functionality without requiring any modifications to the existing legitimate code. As such the original bytecode remains intact allowing an app evade detection. Poeplau et al. [40] defined several techniques to load external code on a device: with the help of class loaders that allow to extract classes from files in arbitrary locations, through the package context that allow to access resources of other apps, through the use of

native code, with the help of runtime method exec that gives access to a system shell, and through less stealthy installation of .apk files requested by a main .apk file.

- *Exposure*: The typical attacks vectors seen in traditional platforms also exist in mobile environments. However, the specificity of mobile devices opened up new avenues for compromise. As such infection methods usually not monitored by traditional IDS, e.g., through SMS, MMS, app' markets, have recently gained a wide popularity.
- *Privacy*: As opposed to traditional networks where privacy leakage typically constitutes a small portions of potential threats, private information theft is rapidly becoming one of the major concerns for mobile devices [32, 44].

4 Evaluation Methodology

To investigate evaluation practices employed by security community, we conducted a survey of research work in the area of intrusion detection for smartphones published since the official release of Android platform in 2008. To avoid selection bias, we collected all research papers indexed by Google Scholar for the reviewed time period from 2008 until 2013. This included studies introducing defence mechanisms specifically developed for the smartphone platforms.

Our research study excluded short papers, extended abstracts, non-peer-reviewed research, and papers not available in the English language. To narrow our focus, we further selected research work relevant to intrusion detection; thus, any methods specifically developed for fault detection, safety analysis, etc. were excluded. The final set of 120 papers, containing 17 journal and 103 conference/workshop papers, was reviewed manually without use of any automatic search techniques. Each of the selected papers were put through at least two evaluation rounds to reduce classification errors.

5 Overview of the Reviewed Studies

In the past several years the number of studies in the field of mobile phone security has been steadily increasing as our survey shows in Figures 1, 2.

Even though most of the research focus on Android, we have seen other platforms used as testing platform as illustrated in Table 1. Traditionally, there have been various classifications relating to intrusion detection mechanisms. Statistics about the surveyed papers with regards to these classifications are

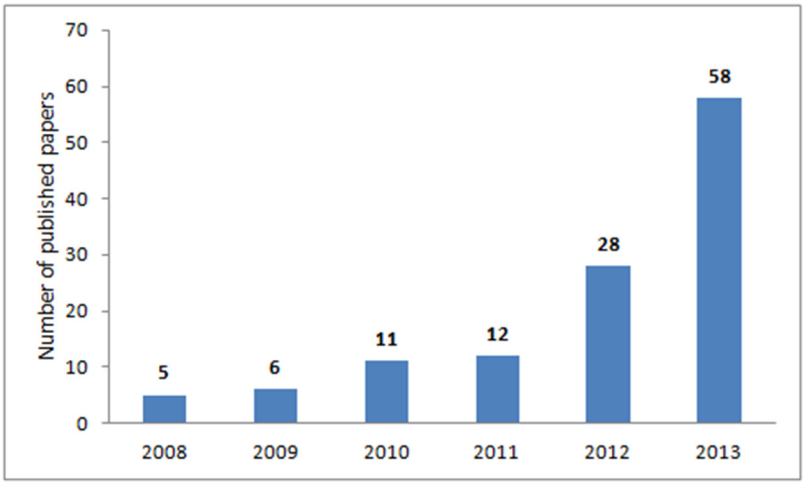


Figure 1 The reviewed papers: a perspective over the years

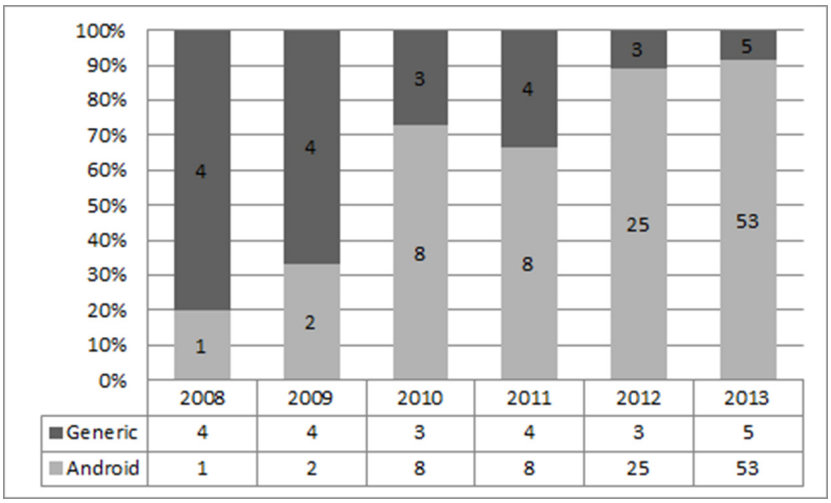


Figure 2 The reviewed papers: a perspective over the years

Table 1 The details of generic approaches

Testing Platforms	
Symbian OS	35% (8 papers out of 23)
Windows Mobile OS	26% (6 papers out of 23)
Hybrid	39% (9 papers out of 23)

Table 2 The details of the surveyed papers

Papers by intrusion detection types	
Network-Based methods	5% (6 papers out of 120)
Hybrid Methods	13% (15 papers out of 120)
Host-Based Methods	82% (99 papers out of 120)
Application level	64% (73 papers out of 114*)
Operating system level	3% (4 papers out of 114)
Hardware level	4% (5 papers out of 114)
Hybrid	29% (32 papers out of 114)
Malware Detection	67% (80 papers out of 120)
By detection approach:	
Anomaly-Based	55% (44 papers out of 80)
Signature-Based	44% (35 papers out of 80)
Hybrid	1% (1 papers out of 80)
By focus:	
Malicious Apps	65% (52 papers out of 80)
Information Leakage	19% (15 papers out of 80)
System Behavior	16% (13 papers out of 80)
Papers by applied detection approach	
Anomaly-Based	58% (70 papers out of 120)
Signature-Based	40% (48 papers out of 120)
Hybrid	2% (2 papers out of 120)
Papers by a level of invasiveness	
Static	35% (42 papers out of 120)
Dynamic	48% (57 papers out of 120)
Hybrid	17% (21 papers out of 120)

shown in Table 2. Among the reviewed papers, the majority of the studies focused on a host-based detection (99 papers out of 120), with only six papers introducing network-based mechanisms and 15 papers proposing hybrid approaches. In addition to these common categories, we noticed an increased interest in specialized mechanisms for mobile malware detection: 80 papers out of 120 considered various aspects of malware detection either through detection of malicious apps (52 out of 80 papers), detection of information leakage (15 out of 80 papers) or suspicious system behavior (13 out of 80 papers).

5.1 Intrusion Detection Focus

To provide a broad overview of research conducted in the field, we categorize the research based on three common parameters: monitoring scope of the proposed technique, underlying detection approach and level of

*114 papers include 99 pure host-based analysis and additional 15 hybrid approaches.

technique invasiveness. Figure 3 illustrates a high-level summary of these categorizations.

By scope of monitoring In general, suspicious activity can be spotted at least at one of the following levels: application, operating system (OS) or hardware. Events at the application level are generally related to user activity and are often seen through suspicious SMS/MMS messages, unusual keystroke dynamics, etc. Since this is a high-level behavior that is not necessarily exhibited by all malware threats, a placement of detection mechanisms at this level provides only limited coverage. OS level activity, on the other hand, includes events triggered by the built-in OS mechanisms (e.g., system calls) thus giving a better picture of the underlying system behavior. While this is often a preferred location for intrusion detection mechanisms in traditional computer-based environments, it presents a number of problems for resource-constrained mobile phone environments that can only afford lightweight detection techniques. Finally, intrusion detection at hardware level allows researchers to obtain basic measurements of the monitored systems (e.g., CPU, network usage, power supply) that might be indicative of abnormal device behavior, especially when they are compared to normal device usage.

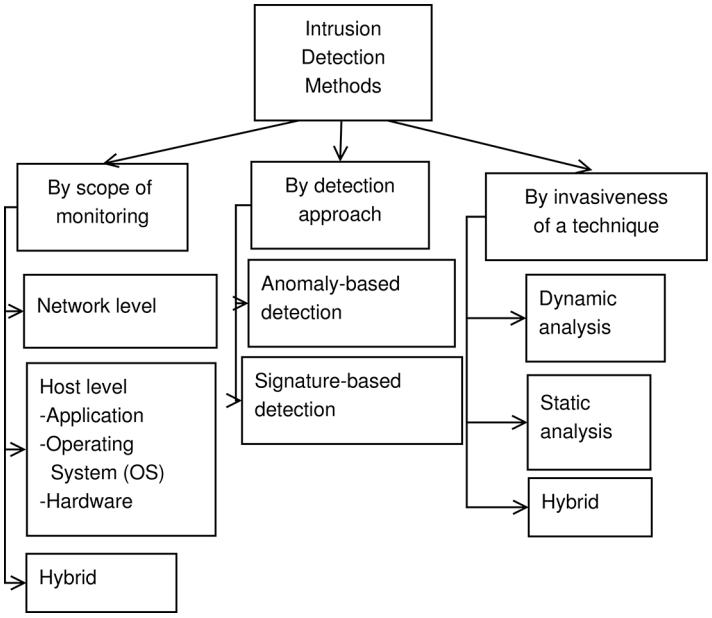


Figure 3 Overview of intrusion detection methods classification

Detection at this level provides a number of benefits that are particularly valuable for mobile devices, including fast, reliable, and scalable intrusion detection. Of the reviewed papers, the majority of studies focus on application level (64%), then OS level (only 3%), and then hardware level (4%). The remaining papers address hybrid levels (29%).

By detection approach Another common classification of intrusion detection techniques, based on how intrusive activities are detected, broadly groups techniques into anomaly-based and signature-based approaches. Since anomaly-detection is perceived as more powerful due to its higher potential to address new threats, it is not surprising to see the majority of the reviewed studies employing it (70 papers out of 120).

Interestingly, a large portion of the reviewed studies is focused on the application of signature-based detection. In spite of criticism of this approach in academic research (due to a high reliance on ever-growing signature bases and the lack of detection of 'zero-day' threats) almost 40% of the reviewed studies (48 out of 120) employed signature-based detection.

By invasiveness of technique The approaches for detection of intrusive activity can be further broken down into dynamic (behavior-based) and static detection, depending on the level of invasiveness of the intrusion detection system. Dynamic techniques generally adopt a black box approach by focusing on behavior of a target system in general or specific files/apps in particular. Since this detection is only possible during the execution of a program, it has a limited focus and is often at the mercy of the executable malware that might or might not exhibit suspicious behavior in a given running environment. Static detection, on the other hand, allows the researcher to analyze an app/file without its execution (via disassembly or source code analysis) and is thus considered to be more thorough, although it is more expensive. Among the reviewed studies, 48% of the papers employed dynamic analysis, with 35% focusing on static detection.

5.2 Trends

The vast popularity of mobile phones, and specifically the Android platform resulted in a rapid increase of mobile malware. One of the major sources of malware became third party app markets [29, 52]. For example, due to popularity of Android platform, we have seen a arise of alternative Android app markets (i.e. not officially supported by Google), often known for their lack of security checks and thus favored by malware writers. Figure 4 clearly shows this trend. Although the use of Google Play market slightly increased

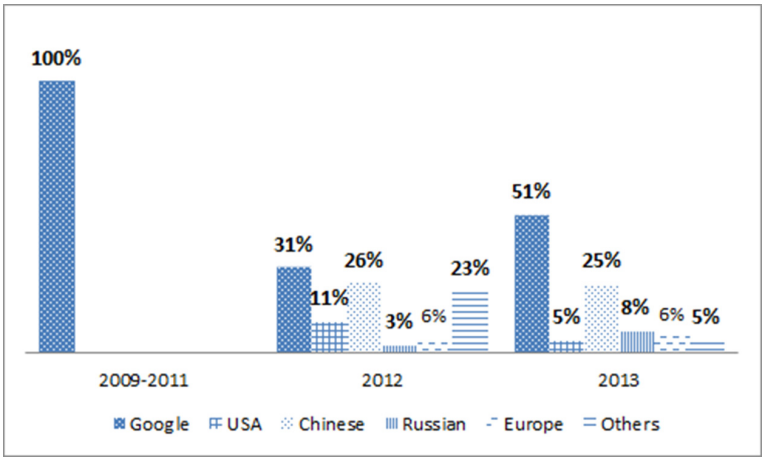
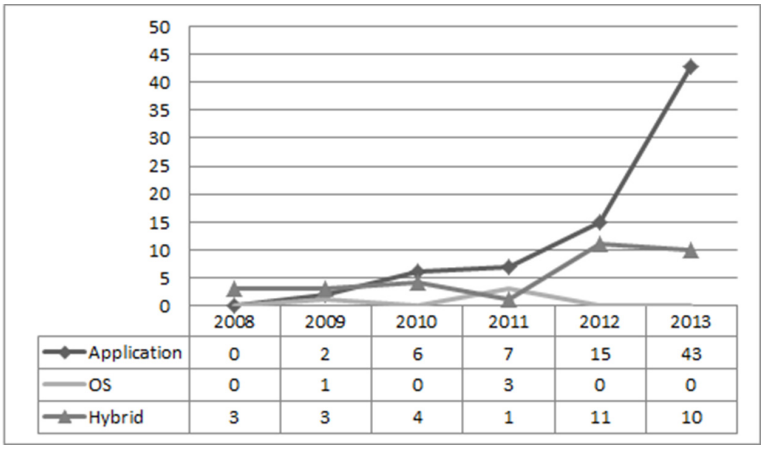


Figure 4 The use of apps from various markets in the surveyed works



	2008	2009	2010	2011	2012	2013
Application	0	2	6	7	15	43
OS	0	1	0	3	0	0
Hybrid	3	3	4	1	11	10

Figure 5 Trends of Research Focused

from 2012 to 2013, it has a significantly smaller share compared to the newly-emerging markets.

Among other trends noticed in the reviewed studies is a clear tendency towards application-level detection (Figure 5). The interest in application-level detection has been steadily increasing and in 2013 the proportion of studies with this scope reached 64%.

6 Detailed Survey Observations

The impact of data quality on the outcomes of the various studies was emphasized throughout the research. Indeed, the ability of a study to predict the performance of a method in a real deployment environment is highly dependent on the ability of a dataset to reflect conditions of that deployment setting.

Datasets. Although the mobile phone setting is not an exception, the lack of comprehensive datasets (mostly attributed to immaturity of the field) has posed a significant problem [52]. There have been several attempts to create structured, comprehensive datasets specifically for evaluating mobile security solutions. Table 3 lists some of these publicly available resources. In spite of their variety, most of these sets have limited applicability and/or require additional preprocessing. The lack of appropriate datasets is reiterated throughout the papers in our survey: all papers that involved experimental studies (110 out of 120 papers) employed self-constructed datasets (see table 4). Although the researchers' intentions are understandable, the lack of standardized datasets employed across the majority of studies raises several concerns.

The first concern is the transparency of a dataset. Customized datasets require collecting data, preprocessing activities (e.g., anonymization, cleaning out redundant data, assuring necessary balance of normal and suspicious behavior), and validating (e.g., confirmation of 'ground truth'). These activities are tedious, time-consuming and often require specialized infrastructure, including equipment and necessary permissions for collecting private data. Unless these activities are fully described and the dataset made available, there is little assurance that a dataset will be representative of a deployment environment. Although the initial step can be avoided with the use of malware repositories, the rest of the concerns remain.

Due to the nature of smartphone threats, the authors of all 110 papers that conducted an experimental study employed a set of mobile apps for this analysis. Out of these 110 papers, 91 papers used real apps collected through app markets and online mobile repositories, three papers implemented their own version of apps and five papers used a combination of these two categories (real and self-written apps).

The majority of the real apps used in surveyed papers were collected through various Android markets (58% of papers) with the assumption that apps coming from a market are benign. With the increasing number of reports showing markets being adopted by hackers as one of the ways for malware distribution, it is expected that all market apps in any experimentation study

Table 3 The existing sources of data for evaluation of mobile security solutions

Name	Institution	Description	Access	Size	Fixed	Labeled	Year	Platform
MIT Reality mining dataset [23]	MIT Human Dynamics Lab	Data collected on smartphones on users' location, communication and device usage behavior for over 9 months; includes only normal user behavior.	Free registration	–	Yes	Yes	2005	Collected on Nokia platform
Android Malware Genome Project [52]	North Carolina State University	Collected malware samples from 49 representative Android families.	Academic registration	1260 samples	Yes	Yes	2011	Android
Mobile Malware in the wild dataset [27]	University of California, Berkeley	Collected malware in the wild from 2009 to 2011	Public Information, not samples.	46 samples	Yes	Yes	2011	iOS, Symbian, Android
Contagio mobile [4]	Personal	Digital place to share mobile malware samples and Analysis, based on a centralized community effort.	Public	330 samples	No	Yes	2011	All mobile
Google group mobile. malware [8]	Community effort	A mailing list for researchers in mobile malware field. Contains material related to new mobile malware samples, analysis, new techniques, questions pertaining to the field, and other related data.	Free registration	–	No	Partially	2010	iOS, Android, Symbian, Windows mobile

AndroMalShare [1]	Xi'an Jiaotong University	This project focuses on sharing Android malware samples. A sampled scanned by SandDrois and provides a detailed report and detection results from several anti-virus tools.	Academic registration	8745 samples	No	Yes	2013	Android
VirusShare [10]	Personal	A personal effort to share samples collected over the years, and accepting contributions from others.	By invitation	–	No	Partially	2011	All type, including mobile
VirusTotal Malware Intelligence Services [11]	Google subsidiary	Free online service for analysis of suspicious binaries and URLs.	Public, only analysis	–	No	Yes	2002	All type, including mobile
SMS Spam Collection [9]	Personal	Collection of SMS messages used for mobile phone spam studies collected by different research groups and by crawling the Internet.	Public	5,574 messages	Yes	Yes	2006–2007	Mobile phones
Android Malware Performance Counter Data [21]	CASTL, Columbia University	Collection of performance counters for Android.	Free registration	–	Yes	Yes	2012	Android ARM

Table 3 Continued

Name	Institution	Description	Access	Size	Fixed	Labeled	Year	Platform
Mining Permission Request Patterns [28]	University of California, Berkeley	The dataset provides the permissions requested by the application., the price, the number of downloads, the average user rating, and a short prosaic description	Public	188,389	Yes	Yes	2011	Android
DroidAnalytics [51]	ANSRLab, Chinese University of Hong Kong	Android malware from 102 different families, with 342 of them being zero-day malware samples from six different families.	Academic registration	2494	Yes	Yes	2013	Android
Drebin [15]	University of Gottingen	Android malware from 176 different families.	Academic registration	5560	Yes	Yes	2010–2012	Android

Table 4 Experimentation practices employed in the surveyed works

Datasets	
Data collection detail:	
Market-Based	52% (63 papers out of 120)
Host-Based	43% (51 papers out of 120)
Network-Based	5% (6 papers out of 120)
Datasets details:	
Self-constructed set	97%(107 papers out of 110)
MIT Reality	3%(3 papers out of 110)
Dataset sharing	3% (3 papers out of 110)
Real apps	83% (91 papers out of 110)
Self-written	11% (12 papers out of 110)
Hybrid	6% (7 papers out of 110)
Normal apps:	
From Market	54% (53 papers out of 98)
Not specified	46% (45 papers out of 98)
Malware samples:	
From Repositories	16% (16 papers out of 98)
From Known Data set	17% (17 papers out of 98)
Hybrid	2% (2 papers out of 98)
Not specified	65% (63 papers out of 98)
Evaluation	
Performed experimental study	90% (108 papers out of 120)
Used simulation	2% (2 papers out of 120)
Do not perform experimental study	8% (10 papers out of 120)
Among 110 that performed experiments:	
Reported evaluation results	93% (102 papers out of 110)
Compared with other techniques	15% (17 papers out of 102)
Involved Internet Access	15% (31 papers out of 78*)
Used Monkey tool	15% (11 papers out of 78)
Employed evaluation metrics:	
Detection Rate	23% (25 papers out of 110)
FPR	35% (39 papers out of 110)
Recall	26% (29 papers out of 110)
Precision	10% (11 papers out of 110)
ROC curves	11% (12 papers out of 110)
AUC	12% (13 papers out of 110)
Self-developed metrics	11% (12 papers out of 110)
DR or FPR or ROC Curves	45% (50 papers out of 110)

will undergo a thorough check to ensure their legitimacy. However, among these studies only 27 verified that apps are malware free.

*78 papers include 57 dynamic analysis and additional 21 hybrid methods.

The other concern related to the use of several sources (i.e., markets) for compiling a single dataset is the necessity of data cleaning. Since many authors distribute apps through a number of markets, duplicative apps (both legitimate and malware) are commonly encountered in different markets. In the reviewed set only 27% of the papers reported at least some activity related to data cleaning.

Although malware apps can be obtained from a number of sources, only 16 out of 98[1] papers utilized existing mobile malware repositories, while several papers reported the use of self-implemented malware apps. With the appearance of research-based mobile datasets, several studies have turned their attention to already prepared data. As such, 17 out of 98 papers used samples from existing malware sets. However, all of these studies have only partially used the datasets, either removing or complementing the existing samples with additional data. This move is another indication of a dire lack of suitable datasets in mobile phone security.

Table 5 presents a summary of the distribution of legitimate and malicious apps in the employed datasets. Throughout our analysis, we observed a very large differences in the sizes of employed datasets. The main concern that this variability raises is the presence of studies with a very small number of samples. Given the availability of data, especially in recent years, evaluations performed on only three malware samples is hardly representative and thus unjustifiable.

The second concern relates to the lack of standardized datasets available to researchers. Among the reviewed studies only three papers made datasets publicly available[2].

Table 5 The sizes of the employed data set in the surveyed papers

Years	Total Size		Normal Samples		Malware Samples	
	Smallest	Largest	Smallest	Largest	Smallest	Largest
2013	8	276016	3	150368	3	12158
2012	6	482514	20	207865	5	378
2011	5	42000	2	13098	3	32
2010	1	2285	30	2285	1	5
2009	240	311	311	311	2	240
2008	4	1000	3	1000	4	7
All years	1	482514	3	207865	1	12158

[1]The 98 papers include 91 studies employing real apps and seven studies using hybrid datasets.

[2]An explicit note about this availability was made in the content of the paper.

The third concern relates to the feasibility of the comparative analysis. Such variability of customized data makes comparative analysis of techniques challenging. Indeed, among the surveyed papers only 15% (17 out of 102 papers) attempted to compare the effectiveness of the developed approach with other methods.

The fourth concern and one of the primary ones when selecting suitable datasets for experiment, is the selection of features that would serve as a basis for analysis. During our survey we extracted 188 unique features used in the reviewed papers. All the papers, even the ones that did not involve the performance of experiments, propose features for analysis. These features were classified according to the categorizations outlined in Figure 6. The statistics for the most commonly employed features are given in Table 6. In spite of 'permissions' being the most commonly used feature throughout the last five years (29% of papers), before that (prior to 2011) 'system calls' was the most popular feature (as illustrated in nine papers out of 22 papers published during 2008–2010). The variability of features in the surveyed research is yet another indication of the immaturity of the field.

Experimentation. Evaluation of the proposed approach is an essential component of any research study in the domain of intrusion detection. Among the reviewed papers, 10 (8%) did not involve any experimentation, while one paper limited its analysis to simulation only. The limitations of simulation in security domain have been repeatedly emphasized in academic studies [13, 39]. Seen as non-exhaustive evaluation technique, simulation does not provide necessary depth of the analysis mostly due to inability to

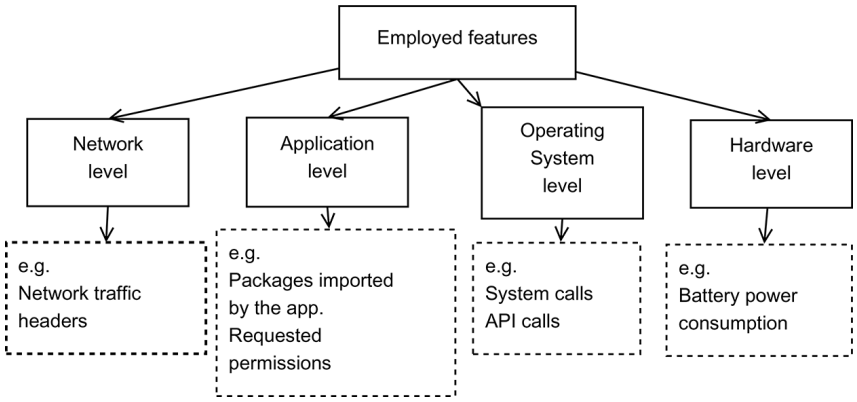


Figure 6 A classification of features

Table 6 Features employed in the surveyed papers

Application level:		
Requested permissions	29%	(69 papers out of 120)
Imported Package	3%	(8 papers out of 120)
Extra information	6%	(14 papers out of 120)
Operating System level:		
API calls	16%	(38 papers out of 120)
System calls	19%	(31 papers out of 120)
Control flow graphs	1%	(3 papers out of 120)
Dataflow	1%	(2 papers out of 120)
Logged Behavior sequences	8%	(19 papers out of 120)
Instructions (Opcode)	6%	(14 papers out of 120)
Hardware level:		
Power consumption	6%	(17 papers out of 120)
Network level level:		
Network traffic	11%	(27 papers out of 120)

guarantee security properties. Constrained to a given scenario simulation does not give insight into a method's performance in unrestricted threat environments or with undiscovered attacks. Although a thorough simulation can provide a sense of average performance of the evaluated technique, it should not be utilized for comprehensive evaluation of security properties.

Among the studies that lacked experiments, only one work reported a proof-of-concept implementation with no mentioning of results. Two of these studies gave a theoretical analysis of the introduced method's performance. Close analysis of the rest of the studies revealed that while all of them discussed some strategies for implementation and potential analysis, none of them offered neither of these.

The details of experimental setup and the employed methodology are necessary to ensure repeatability of the experiments and therefore to facilitate the comparison between techniques. However, as our survey shows most of the researchers neglect to include these details in the study description.

For example, user interface interactions, as one of the triggers of malicious behavior, are known to be essential for analysis of mobile malware [50]. However, among 78 papers that performed dynamic analysis and hybrid analysis of malware, only 11 papers reported the use of user interface interactions to produce events. All of these studies however employed the use of Monkey tool that produces pseudo-random streams of user and system events [14]. Meant for general stress testing of Android apps, the tool is limited to some subset

Table 7 The details of published techniques

Compared with other techniques	
Virus total scanner [11]	23% (4 papers out of 17)
TaintDroid [25]	17% (3 papers out of 17)
Kirin [26]	12% (2 papers out of 17)
Andromaly [45]	12% (2 papers out of 17)
VirusMeter [36]	6% (1 papers out of 17)
Andrubis [35]	6% (1 papers out of 17)
DNADroid [20]	6% (1 papers out of 17)
Androguard [22]	6% (1 papers out of 17)
Clam-AV Scanner [31]	6% (1 papers out of 17)
Manual Analysis	6% (1 papers out of 17)

of actions that do not correctly represent a realistic user of system behavioral patterns.

Similarly, although the access to the Internet is one of the important triggers for many malware, we found that only 31 out of 78 studies reported the use of Internet access in their experiments.

Evaluations. Among the 110 papers that undertook evaluation, eight studies did not report results. Of the 110 studies overall, most works (102) offered experimentation results, which varied significantly from extensive discussion reasoning behind the obtained numbers to a brief mentioning of whether an attack was detected or not.

The evaluation of the method's effectiveness is generally performed along two directions: (1) evaluation of method's overhead in terms of CPU occupancy, memory usage, power consumption and detection speed; and (2) evaluation of the method's detection capability. While both types of experiments are necessary to assess the quality of the newly developed approach, the majority of the surveyed studies are mostly focused on the evaluation of classification accuracy of a method. To assess this accuracy though, it is necessary to compare the new method's performance against known benchmarks in the field. However, due to the infancy of the field such benchmarks are mostly non-existent. As such most of the studies either do not provide any comparison or look at previously proposed methods. Of these surveyed papers, only 17 (15%) compared experimentation results with perviously published intrusion detection techniques and tools. The list of these techniques is given in Table 7. The fact that these techniques are selected for comparison might be an indication of them becoming future benchmarks.

7 Guidelines

The evaluation of intrusion detection methods is an inherently difficult problem due to a variety of factors, from a diversity of monitored events to the unpredictable nature of a deployed environment. Aggravated by the immaturity of the field, evaluation of intrusion detection solutions for smartphone platforms are often guided by subjective choices and individual preferences of the researchers. A review of the experimentation practices employed in this field suggests several important points. Based on the results of our review we formulated several guidelines for proper evaluation of intrusion detection methods for smartphones.

The resulting criteria is offered in three general dimensions of scientific experimentation, formulated by previous research [47, 43]: factors related to *the employed datasets*, *the performed experiments*, and *the performance evaluation*. Since the previous studies have attempted to outline some of the limitations and constraints of scientific experimentation in computer security, we aimed to devise comprehensive guidelines and recommendations for a smartphone setting. The following guidelines can be used as a structure for experimental design and subsequently in manuscript organization:

Employed datasets define applicability and deployability of the developed technique to a real-life setting. To ensure the effectiveness of these qualities the dataset description is an essential component of any manuscript. Although this description will vary depending on the source of data (self-constructed or publicly available), it should provide enough detail to allow an intelligent analysis of the proposed technique. Specifically, the following aspects of employed data should be addressed:

- Data overview:
    - the source of the data, (i.e., whether the dataset is public, proprietary, or self-constructed).
    - quantitative description of malicious and normal events in a dataset (e.g., apps, malware, network flows).
    - how these malicious and normal events were obtained (i.e., simulated, implemented, collected). This should include description of the environment/process. For example, assuming the dataset includes Android apps that are collected from various sources: are these apps free or paid, what are the sources (e.g., list app markets), what malware families these apps represent, why these malware apps are selected while others are excluded? On the other hand, if

the data is collected on the network, then the description should include the duration/time of the collection, access to the Internet, use of visualization, etc.
    - monitored features.
- Data validation:
    - the ground truth established, (i.e., how the legitimacy and abnormality of data is verified). For example, in the case of Android apps, whether or not the collected apps checked through third-party sources.
    - obsolete data removed, (i.e., in the case where a dataset containing malware or apps is employed, sinkholed or inactive samples should be removed).
- Training/testing data:
    - quantitative division of dataset into training and testing sets, corresponding numerical estimation of normal and malicious samples in each.
- Data sharing:
    - data employed in the experiments should be archived for reference for future authors. Even if datasets are not made public, the possibility of sharing on demand should be explicitly indicated in a manuscript.

The performed experiments refer to the setting of the performed experiments and thus allow for transparency of the experiments to be ensured, (i.e., that the conducted experiments are repeatable and comprehensible). Several aspects need to be addressed here to allow for objective evaluation of a study:

- Environmental setting:
    - experimental setup, (i.e., simulation, emulation, experimentation).
    - context for execution for both victim and attack side, including hardware (e.g., devices employed, their topology) and software (e.g., operating systems, NAT, privileges). This description might be complemented with a diagram to avoid ambiguities in interpreting an environmental setup.
    - the employed tools, with an indication of their releases and versions, and parameters.
    - the use of technology (e.g., access to the Internet).

- Experiments:
  - methodology, (i.e., steps involved, allowed user interaction, duration, number of repetitions).
  - the use of sampling, the employed algorithm and justification.

Performance evaluation. The primary goal of this section is to give insight into a proposed method for performance, which is called for a detailed and, more importantly, objective analysis of evaluation.

- Evaluation Methodology:
  - the scripts or procedures used for analysis should be specified. Often a closer analysis of intrusion detection results call for a manual examination of traces. Regardless of the findings this should be stated.
  - the employed statistical apparatus.
- Evaluation metrics:
  - define evaluation metrics. To avoid ambiguities in metric interpretation, a clear definition of a chosen metric should be provided. This becomes critically important when non-traditional metrics are employed, as the lack of consistent terminology hinders researcher's ability to properly identify and apply common methods. As the use of self-developed metrics becomes more widespread, a metric definition should be followed by a clear validation of the proposed metric for a given task.
  - ensure a proper combination of metrics. Detection rate (DR) and false positive rate (FPR) (or their graphical representation ROC curve) are the most widely used metrics in spite of criticism. Several studies have shown that the ROC curve alone or DR/FPR metrics combination might be misleading or simply incomplete for understanding the strengths and weaknesses of intrusion detection systems [18, 48, 38, 16, 34]. Although such misinterpretation can be partially avoided when DR/FPR values are complemented with the absolute numbers of false alarms or missed detections, exclusive evaluation of an approach with these methods may not accurately represent the performance of the system.
- Findings:
  - provide numerical results. The reporting of numerical results should be comprehensive (i.e., use complementary metrics that show all

perspectives of the evaluated method performance) and consistent (i.e., reported in the same equivalents). For example, only stating the accuracy of the proposed approach (i.e., that shows the percentage of correctly classified events) gives almost no insight into a method's performance with respect to malicious events. On the other hand, for example, stating detection rates in percentages, while indicating false positive rates in absolute numbers, makes it challenging for a researcher to interpret the numbers.

- interpret the results. The intrusion detection community has traditionally focused on understanding the limitations of detection methods. Therefore, simply stating numbers is not sufficient for interpreting the performance of the detection method. Numerical results should be included to show the soundness of the approach and allow for future comparative analysis of the detection methods, but they are not the main goal. Thus, a close and often manual examination of results (e.g., false positives and false negatives) is necessary to understand and explain to the research community why and when a system performs in a given way.

- Comparison:
  - a comparative analysis of the proposed scheme with the established benchmarks in the field is an essential component of a study. Whenever existing benchmarks are not available an attempt should be made to perform a quantitative comparison against prior results.

8 Conclusion

With the popularity of Android platform, the amount of research studies on security of smartphones is rapidly increasing. While an increasing volume of studies in the field is generally seen as an indicator of a field evolution, the true value of existing work and its impact on the smartphone security progress remains unclear.

As such in spite of variability of tools proposed only a few of them have been accepted by a security community. Among them is an Android analysis tool, Androguard [22]. Effective against even obfuscation techniques, Androguard has been adopted many academic and industry malware analysis solutions, such as Virustotal portal [11], APKInspector [3], Marvinsafe [7], Anubis (Andrubis) [35], Androwarn [2], googleplay-api [5] and MalloDroid [6]. In academic studies, as our review shows, there is less

consensus and studies employ a variability of tools for evaluation of new solutions.

The validity of experimental research in academic computer science in general has shown to be questionable. Through our review we also confirmed our initial hypothesis, showing that immaturity of the smartphone security negatively affects experimentation practices adopted in the field. In this context, it is plausible to suggest that the lack of adoption of the developed innovations by industry can be attributed to the lack of proper rigor in experimentation that shadows the true value of the developed solutions. Among the factors that contribute to the lack of scientific rigor in experimentation are the lack of consistency and transparency in the use of datasets, the lack of clear understanding of relevant features, the lack of benchmarked experimentation, and the biased selection of malicious apps for analysis.

While the infancy of the field of mobile phone security might justify some shortcomings in experimentation, many of the pitfalls can be avoided with proper practices established and adopted by a security community. To facilitate acceptance of this practice, we formulated a set of guidelines for a proper evaluation of intrusion detection methods for smartphone Platforms. The suggested guidelines allow a detailed description and analysis of experimentation results, and, as such, might be used for designing experimentation studies as well as for structuring experimentation sections in a manuscript. While some of the suggested guidelines might be seen as common sense, we believe that framing them in a structured way will help researchers to improve experimentation practices by providing them with a methodological reference.

9 Acknowledgment

The first author graciously acknowledges the funding from University of Hail at Saudi Arabia.

References

[1] Andromalshare. http://202.117.54.231:8080/.
[2] Androwarn. https://github.com/maaaaz/androwarn.
[3] Apkinspector. https://github.com/honeynet/apkinspector/.
[4] contagio mobile. http://contagiodump.blogspot.ca/.
[5] googleplay-api. http://www.segmentationfault.fr/publications/reversing-google-play- and-micro-protobuf-applications/.
[6] Mallodroid. http://www2.dcsec.uni-hannover.de/files/android/p50-fahl.pdf.
[7] Marvinsafe. http://www.marvinsafe.com/.
[8] Mobile malware forum. https://groups.google.com/forum/?fromgroups=#!forum/mobilemalware.
[9] Sms spam collection v.1. http://www.dt.fee.unicamp.br/tiago/smsspamcollection/.
[10] Virusshare.com - because sharing is caring. http://virusshare.com/.
[11] Virustotal malware intelligence services. https://www.virustotal.com.
[12] A. Amamra, C. Talhi, and J.-M. Robert. Smartphone malware detection: From a survey towards taxonomy. In *Malicious and Unwanted Software (MALWARE), 2012 7th International Conference on*. IEEE, 2012.
[13] T. R. Andel and A. Yasinac. On the credibility of manet simulations. *Computer*, 39 (7): 48–54, July 2006.
[14] Android Developers. Ui/application exerciser monkey. http://developer.android.com/tools/help/monkey.html.
[15] D. Arp, M. Spreitzenbarth, M. Huübner, H. Gascon, K. Rieck, and C. Siemens. Drebin: Effective and explainable detection of android malware in your pocket. 2014.
[16] S. Axelsson. The base-rate fallacy and its implications for the difficulty of intrusion detection. In *CCS '99: Proceedings of the 6th ACM conference on Computer and communications security*, pages 1–7, New York, NY, USA, 1999. ACM.
[17] M. Becher, F. Freiling, J. Hoffmann, T. Holz, S. Uellenbeck, and C. Wolf. Mobile security catching up? revealing the nuts and bolts of the security of mobile devices. In *Security and Privacy (SP), 2011 IEEE Symposium on*, pages 96–111, 2011.
[18] A. A. Cárdenas, J. S. Baras, and K. Seamon. A framework for the evaluation of intrusion detection systems. In *SP '06: Proceedings of the 2006 IEEE Symposium on Security and Privacy*, pages 63–77, Washington, DC, USA, 2006. IEEE Computer Society.

[19] M. Chandramohan and H. B. K. Tan. Detection of mobile malware in the wild. *Computer*, 45(9): 65–71, 2012.

[20] J. Crussell, C. Gibler, and H. Chen. Attack of the clones: Detecting cloned applications on android markets. In *Computer Security–ESORICS 2012*, pages 37–54. Springer, 2012.

[21] J. Demme, M. Maycock, J. Schmitz, A. Tang, A. Waksman, S. Sethumadhavan, and S. Stolfo. On the feasibility of online malware detection with performance counters. In *Proceedings of the 40th annual international symposium on Computer architecture*, ISCA '13, New York, NY, USA, 2013. ACM.

[22] A. Desnos and G. Gueguen. Android: From reversing to decompilation. In *Blackhat*, 2011.

[23] N. Eagle and A. (Sandy) Pentland. Reality mining: sensing complex social systems. *Personal and Ubiquitous Computing*, 10(4): 255–268, 2006.

[24] W. Enck. Defending users against smartphone apps: techniques and future directions. In *Proceedings of the 7th international conference on Information Systems Security*, ICISS'11, pages 49–70, Berlin, Heidelberg, 2011. Springer-Verlag.

[25] W. Enck, P. Gilbert, B.-G. Chun, L. P. Cox, J. Jung, P. McDaniel, and A. Sheth. Taintdroid: An information-flow tracking system for real-time privacy monitoring on smartphones. In *OSDI*, volume 10, pages 1–6, 2010.

[26] W. Enck, M. Ongtang, and P. McDaniel. On lightweight mobile phone application certification. In *Proceedings of the 16th ACM conference on Computer and communications security*, pages 235–245. ACM, 2009.

[27] A. P. Felt, M. Finifter, E. Chin, S. Hanna, and D. Wagner. A survey of mobile malware in the wild. In *Proceedings of the 1st ACM workshop on Security and privacy in smartphones and mobile devices*, SPSM '11, New York, NY, USA, 2011. ACM.

[28] M. Frank, B. Dong, A. P. Felt, and D. Song. Mining permission request patterns from android and facebook applications. pages 870–875, 12 2012.

[29] G. Kelly. Report: 97% of mobile malware is on android. this is the easy way you stay safe. http://www.forbes.com/sites/gordonkelly/2014/03/24/report-97-of-mobile-malware-is-on-android-this-is-the-easy-way-you-stay-safe/. Accessed on Mar 2014.

[30] L. Ketari and M. A. Khanum. A review of malicious code detection techniques for mobile devices. *International Journal of Computer Theory and Engineering*, 4(2): 212–216, 2012.

[31] T. Kojm. Clamav anti-virus. http: //www.clamav.net/.

[32] B. Krishnamurthy and C. E. Wills. Privacy leakage in mobile online social networks. In *Proceedings of the 3rd Conference on Online Social Networks*, WOSN'10, pages 4–4, Berkeley, CA, USA, 2010. USENIX Association.

[33] M. La Polla, F. Martinelli, and D. Sgandurra. A survey on security for mobile devices. *Communications Surveys Tutorials, IEEE*, 15, 2013.

[34] A. Lazarevic, L. Ertoz, V. Kumar, A. Ozgur, and J. Srivastava. A comparative study of anomaly detection schemes in network intrusion detection. *Proceedings of the Third SIAM International Conference on Data Mining*, 2003.

[35] M. Lindorfer. Andrubis: A tool for analyzing unknown android applications. http://blog.iseclab.org/2012/06/04/andrubis-a-tool-for-analyzing-unknown-android-applications-2/.

[36] L. Liu, G. Yan, X. Zhang, and S. Chen. Virusmeter: Preventing your cellphone from spies. In *Recent Advances in Intrusion Detection*, pages 244–264. Springer, 2009.

[37] D. Maslennikov and Y. Namestnikov. Kaspersky security bulletin 2012. the overall statistics for 2012, 2012.

[38] J. McHugh. The 1998 Lincoln Laboratory IDS evaluation. In *RAID '00: Proceedings of the Third International Workshop on Recent Advances in Intrusion Detection*, pages 145–161, London, UK, 2000. Springer-Verlag.

[39] I. Parris, F. Ben Abdesslem, and T. Henderson. Facebook or fakebook? the effects of simulated mobile applications on simulated mobile networks. *Ad Hoc Netw.*, 12: 35–49, Jan. 2014.

[40] S. Poeplau, Y. Fratantonio, A. Bianchi, C. Kruegel, and G. Vigna. Execute this! analyzing unsafe and malicious dynamic code loading in android applications. In *Proceedings of the Network and Distributed System Security Symposium (NDSS)*, 2014.

[41] K. L. G. RESEARCH and A. TEAM. Kaspersky security bulletin 2013. the overall statistics for 2013, 2013.

[42] H. Ringberg, M. Roughan, and J. Rexford. The need for simulation in evaluating anomaly detectors. *SIGCOMM Comput. Commun. Rev.*, 38(1): 55–59, 2008.

[43] C. Rossow, C. J. Dietrich, C. Grier, C. Kreibich, V. Paxson, N. Pohlmann, H. Bos, and M. v. Steen. Prudent practices for designing malware experiments: Status quo and outlook. In *Proceedings of the 2012 IEEE Symposium on Security and Privacy*, SP '12, pages 65–79, Washington, DC, USA, 2012. IEEE Computer Society.

[44] N. Seriot. iphone privacy. In *Proceedings of the Black Hat*, Arlington, Virginia, USA, 2010.

[45] A. Shabtai, U. Kanonov, Y. Elovici, C. Glezer, and Y. Weiss. "andromaly": A behavioral malware detection framework for android devices. *J. Intell. Inf. Syst.*, 38(1): 161–190, 2012.

[46] F. Shahzad, M. A. Akbar, and M. Farooq. A survey on recent advances in malicious applications analysis and detection techniques for smartphones. 2012.

[47] M. Tavallaee, N. Stakhanova, and A. A. Ghorbani. Toward credible evaluation of anomaly-based intrusion-detection methods. *IEEE Transactions on Sys. Man Cyber Part C*, 40(5): 516–524, Sept. 2010.

[48] J. W. Ulvila and J. E. Gaffney. Evaluation of intrusion detection systems. *Journal of Research of the National Institute of Standards and Technology*, 108(6): 453–471, 2003.

[49] Q. Yan, Y. Li, T. Li, and R. Deng. Insights into malware detection and prevention on mobile phones. In *International Conference on Security Technology*, volume 58 of *Communications in Computer and Information Science*, pages 242–249. Springer Berlin Heidelberg, 2009.

[50] C. Zheng, S. Zhu, S. Dai, G. Gu, X. Gong, X. Han, and W. Zou. Smartdroid: An automatic system for revealing ui-based trigger conditions in android applications. In *Proceedings of the Second ACM Workshop on Security and Privacy in Smartphones and Mobile Devices*, SPSM '12, pages 93–104, New York, NY, USA, 2012. ACM.

[51] M. Zheng, M. Sun, and J. Lui. Droid analytics: A signature based analytic system to collect, extract, analyze and associate android malware. In *Trust, Security and Privacy in Computing and Communications (TrustCom), 2013 12th IEEE International Conference on*, pages 163–171. IEEE, 2013.

[52] Y. Zhou and X. Jiang. Dissecting Android malware: Characterization and evolution. In *IEEE Symposium on Security and Privacy (SP)*, pages 95–109. IEEE, 2012.

Appendix

List of the Reviewed Papers

Android Platform:	
Malware Detection	[1, 2, 3, 4, 5, 7, 8, 9, 10, 11, 12, 15, 16, 17, 20, 21, 22, 23, 25, 27, 28, 29, 30, 31, 34, 36, 38, 39, 40, 41, 42, 46, 48, 50, 51, 52, 53, 54, 56, 57, 58, 59, 60, 61, 62, 64, 65, 67, 68, 70, 71, 72, 75, 76, 77, 78, 80, 87, 93, 94, 95, 96, 98, 99, 100, 102, 104, 105, 110, 111, 116]
NIDS-HIDS	[6, 13, 43, 45, 47, 49, 69, 73, 85, 106]
NIDS	[35, 44, 66, 9]
HIDS	[14, 26, 32, 37, 63, 74, 79, 84, 86, 92, 101, 103]
Symbian OS:	
Malware Detection	[82, 107, 109, 112, 115]
HIDS	[108, 117, 119]
Windows Mobile OS:	
Malware Detection	[83, 89]
NIDS-HIDS	[118, 120]
HIDS	[88, 113]
Other Platforms:	
Malware Detection	[18, 90]
NIDS-HIDS	[24, 33, 55]
NIDS	[19, 81]
HIDS	[91, 114]

References

[1] Sanae Rosen, Zhiyun Qian, and Z Morely Mao. Appprofiler: a flexible method of exposing privacy-related behavior in android applications to end users. In *Proceedings of the third ACM conference on Data and application security and privacy*, pages 221–232. ACM, 2013.

[2] Saurabh Chakradeo, Bradley Reaves, Patrick Traynor, and William Enck. Mast: triage for market-scale mobile malware analysis. In *Proceedings of the sixth ACM conference on Security and privacy in wireless and mobile networks*, pages 13–24. ACM, 2013.

[3] Johannes Hoffmann, Martin Ussath, Thorsten Holz, and Michael Spreitzenbarth. Slicing droids: program slicing for smali code. In *Proceedings of the 28th Annual ACM Symposium on Applied Computing*, pages 1844–1851. ACM, 2013.

[4] Kevin Joshua Abela, Jan Raynier Delas Alas, Don Kristopher Angeles, Robert Joseph Tolentino, and Miguel Alberto Gomez. Automated malware detection for android amda. In *The Second International*

Conference on Cyber Security, Cyber Peacefare and Digital Forensic (CyberSec2013), pages 180–188. The Society of Digital Information and Wireless Communication, 2013.

[5] Zarni Aung and Win Zaw. Permission-based android malware detection. *International Journal of Scientific and Technology Research*, 2(3): 228–234, 2013.

[6] Michael Spreitzenbarth, Felix Freiling, Florian Echtler, Thomas Schreck, and Johannes Hoffmann. Mobile-sandbox: having a deeper look into android applications. In *Proceedings of the 28th Annual ACM Symposium on Applied Computing*, pages 1808–1815. ACM, 2013.

[7] Mo Ghorbanzadeh, Yang Chen, Zhongmin Ma, T Charles Clancy, and Robert McGwier. A neural network approach to category validation of android applications. In *Computing, Networking and Communications (ICNC), 2013 International Conference on*, pages 740–744. IEEE, 2013.

[8] Min Zheng, Mingshen Sun, and John Lui. Droid analytics: A signature based analytic system to collect, extract, analyze and associate android malware. In *Trust, Security and Privacy in Computing and Communications (TrustCom), 2013 12th IEEE International Conference on*, pages 163–171. IEEE, 2013.

[9] Borja Sanz, Igor Santos, Javier Nieves, Carlos Laorden, Inigo Alonso-Gonzalez, and Pablo G Bringas. Mads: Malicious android applications detection through string analysis. In *Network and System Security*, pages 178–191. Springer, 2013.

[10] Yibing Zhongyang, Zhi Xin, Bing Mao, and Li Xie. Droidalarm: an all-sided static analysis tool for android privilege-escalation malware. In *Proceedings of the 8th ACM SIGSAC symposium on Information, computer and communications security*, pages 353–358. ACM, 2013.

[11] Wu Zhou, Yajin Zhou, Michael Grace, Xuxian Jiang, and Shihong Zou. Fast, scalable detection of piggybacked mobile applications. In *Proceedings of the third ACM conference on Data and application security and privacy*, pages 185–196. ACM, 2013.

[12] Vaibhav Rastogi, Yan Chen, and William Enck. Appsplayground: Automatic security analysis of smartphone applications. In *Proceedings of the third ACM conference on Data and application security and privacy*, pages 209–220. ACM, 2013.

[13] Hiroki Kuzuno and Satoshi Tonami. Signature generation for sensitive information leakage in android applications. In *Data Engineering*

Workshops (ICDEW), 2013 IEEE 29th International Conference on, pages 112–119. IEEE, 2013.

[14] Johannes Hoffmann, Stephan Neumann, and Thorsten Holz. Mobile malware detection based on energy fingerprints a dead end? In *Research in Attacks, Intrusions, and Defenses*, pages 348–368. Springer, 2013.

[15] Yousra Aafer, Wenliang Du, and Heng Yin. Droidapiminer: Mining api-level features for robust malware detection in android. In *Security and Privacy in Communication Networks*, pages 86–103. Springer, 2013.

[16] John Demme, Matthew Maycock, Jared Schmitz, Adrian Tang, Adam Waksman, Simha Sethumadhavan, and Salvatore Stolfo. On the feasibility of online malware detection with performance counters. In *Proceedings of the 40th Annual International Symposium on Computer Architecture*, pages 559–570. ACM, 2013.

[17] Hugo Gascon, Fabian Yamaguchi, Daniel Arp, and Konrad Rieck. Structural detection of android malware using embedded call graphs. In *Proceedings of the 2013 ACM workshop on Artificial intelligence and security*, pages 45–54. ACM, 2013.

[18] Yuan Zhang, Min Yang, Bingquan Xu, Zhemin Yang, Guofei Gu, Peng Ning, X Sean Wang, and Binyu Zang. Vetting undesirable behaviors in android apps with permission use analysis. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*, pages 611–622. ACM, 2013.

[19] Ruofan Jin and Bing Wang. Malware detection for mobile devices using software-defined networking. In *Research and Educational Experiment Workshop (GREE), 2013 Second GENI*, pages 81–88. IEEE, 2013.

[20] Hyo-Sik Ham and Mi-Jung Choi. Analysis of android malware detection performance using machine learning classifiers. In *ICT Convergence (ICTC), 2013 International Conference on*, pages 490–495. IEEE, 2013.

[21] Federico Maggi, Andrea Valdi, and Stefano Zanero. Androttotal: a flexible, scalable toolbox and service for testing mobile malware detectors. In *Proceedings of the Third ACM workshop on Security and privacy in smartphones & mobile devices*, pages 49–54. ACM, 2013.

[22] Suleiman Y Yerima, Sakir Sezer, Gavin McWilliams, and Igor Muttik. A new android malware detection approach using bayesian classification. In *Advanced Information Networking and Applications (AINA), 2013 IEEE 27th International Conference on*, pages 121–128. IEEE, 2013.

[23] Brandon Amos, Hamilton Turner, and Jules White. Applying machine learning classifiers to dynamic android malware detection at scale. In *Wireless Communications and Mobile Computing Conference (IWCMC), 2013 9th International*, pages 1666–1671. IEEE, 2013.

[24] Byungha Choi, Sung-Kyo Choi, and Kyungsan Cho. Detection of mobile botnet using vpn. In *Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2013 Seventh International Conference on*, pages 142–148. IEEE, 2013.

[25] Aiman A Abu Samra and Osama A Ghanem. Analysis of clustering technique in android malware detection. In *Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2013 Seventh International Conference on*, pages 729–733. IEEE, 2013.

[26] Bryan Dixon and Shivakant Mishra. Power based malicious code detection techniques for smartphones. In *Trust, Security and Privacy in Computing and Communications (TrustCom), 2013 12th IEEE International Conference on*, pages 142–149. IEEE, 2013.

[27] Parvez Faruki, Vijay Ganmoor, Vijay Laxmi, MS Gaur, and Ammar Bharmal. Androsimilar: robust statistical feature signature for android malware detection. In *Proceedings of the 6th International Conference on Security of Information and Networks*, pages 152–159. ACM, 2013.

[28] Kevin Joshua Abela, Don Kristopher Angeles, Jan Raynier Delas Alas, Robert Joseph Tolentino, and Miguel Alberto Gomez. An automated malware detection system for android using behavior-based analysis amda. *International Journal of Cyber-Security and Digital Forensics (IJCSDF)*, 2(2):1–11, 2013.

[29] Borja Sanz, Igor Santos, Xabier Ugarte-Pedrero, Carlos Laorden, Javier Nieves, and Pablo G Bringas. Instance-based anomaly method for android malware detection. pages 387–394, 2013.

[30] Anand Paturi, Manoj Cherukuri, John Donahue, and Srinivas Mukkamala. Mobile malware visual analytics and similarities of attack toolkits (malware gene analysis). In *Collaboration Technologies and Systems (CTS), 2013 International Conference on*, pages 149–154. IEEE, 2013.

[31] Heqing Huang, Sencun Zhu, Peng Liu, and Dinghao Wu. A framework for evaluating mobile app repackaging detection algorithms. In *Trust and Trustworthy Computing*, pages 169–186. Springer, 2013.

[32] Monica Curti, Alessio Merlo, Mauro Migliardi, and Simone Schiappacasse. Towards energy-aware intrusion detection systems on mobile

devices. In *High Performance Computing and Simulation (HPCS), 2013 International Conference on*, pages 289–296. IEEE, 2013.

[33] Wei Yu, Zhijiang Chen, Guobin Xu, Sixiao Wei, and Nnanna Ekedebe. A threat monitoring system for smart mobiles in enterprise networks. In *Proceedings of the 2013 Research in Adaptive and Convergent Systems*, pages 300–305. ACM, 2013.

[34] Jianlin Xu, Yifan Yu, Zhen Chen, Bin Cao, Wenyu Dong, Yu Guo, and Junwei Cao. Mobsafe: cloud computing based forensic analysis for massive mobile applications using data mining. *Tsinghua Science and Technology*, 18(4), 2013.

[35] Lena Tenenboim-Chekina, Lior Rokach, and Bracha Shapira. Ensemble of feature chains for anomaly detection. In *Multiple Classifier Systems*, pages 295–306. Springer, 2013.

[36] Yibing Zhongyang, Zhi Xin, Bing Mao, and Li Xie. Droidalarm: an all-sided static analysis tool for android privilege-escalation malware. In *Proceedings of the 8th ACM SIGSAC symposium on Information, computer and communications security*, pages 353–358. ACM, 2013.

[37] Hwan-Taek Lee, Minkyu Park, and Seong-Je Cho. Detection and pre vention of lena malware on android. *Journal of Internet Services and Information Security (JISIS)*, 3(3/4):63–71, 2013.

[38] Seung-Hyun Seo, Aditi Gupta, Asmaa Mohamed Sallam, Elisa Bertino, and Kangbin Yim. Detecting mobile malware threats to homeland security through static analysis. *Journal of Network and Computer Applications*, 38:43–53, 2013.

[39] Byeongho Kang, BooJoong Kang, Jungtae Kim, and Eul Gyu Im. Android malware classification method: Dalvik bytecode frequency analysis. In *Proceedings of the 2013 Research in Adaptive and Convergent Systems*, pages 349–350. ACM, 2013.

[40] Suyeon Lee, Jehyun Lee, and Heejo Lee. Screening smartphone applications using behavioral signatures. In *Security and Privacy Protection in Information Processing Systems*, pages 14–27. Springer, 2013.

[41] Parvez Faruki, Vijay Laxmi, Vijay Ganmoor, MS Gaur, and Ammar Bharmal. Droidolytics: Robust feature signature for repackaged android apps on official and third party android markets. In *Advanced Computing, Networking and Security (ADCONS), 2013 2nd International Conference on*, pages 247–252. IEEE, 2013.

[42] Shaoyin Cheng, Shengmei Luo, Zifeng Li, Wei Wang, Yan Wu, and Fan Jiang. Static detection of dangerous behaviors in android apps. In *Cyberspace Safety and Security*, pages 363–376. Springer, 2013.

[43] Zhizhong Wu, Xuehai Zhou, and Jun Xu. A result fusion based distributed anomaly detection system for android smartphones. *Journal of Networks*, 8(2), 2013.

[44] Ryan Johnson, Zhaohui Wang, Angelos Stavrou, and Jeff Voas. Exposing software security and availability risks for commercial mobile devices. In *Reliability and Maintainability Symposium (RAMS), 2013 Proceedings-Annual*, pages 1–7. IEEE, 2013.

[45] K Saritha and R Samaiah. Behavior analysis of mobile system in cloud computing. In *International Journal of Engineering Research and Technology*, volume 2. ESRSA Publications, 2013.

[46] Thomas Eder, Michael Rodler, Dieter Vymazal, and Markus Zeilinger. Ananas-a framework for analyzing android applications. In *Availability, Reliability and Security (ARES), 2013 Eighth International Conference on*, pages 711–719. IEEE, 2013.

[47] Roshanak Roshandel, Payman Arabshahi, and Radha Poovendran. Lidar: a layered intrusion detection and remediationframework for smartphones. In *Proceedings of the 4th international ACM Sigsoft symposium on Architecting critical systems*, pages 27–32. ACM, 2013.

[48] Mohammad Karami, Mohamed Elsabagh, Parnian Najafiborazjani, and Angelos Stavrou. Behavioral analysis of android applications using automated instrumentation. In *Software Security and Reliability-Companion (SERE-C), 2013 IEEE 7th International Conference on*, pages 182–187. IEEE, 2013.

[49] Fangfang Yuan, Lidong Zhai, Yanan Cao, and Li Guo. Research of intrusion detection system on android. In *Services (SERVICES), 203 IEEE Ninth World Congress on*, pages 312–316. IEEE, 2013.

[50] Ryo Sato, Daiki Chiba, and Shigeki Goto. Detecting android malware by analyzing manifest files. *Proceedings of the Asia-Pacific Advanced Network*, 36: 23–31, 2013.

[51] Dong-uk Kim, Jeongtae Kim, and Sehun Kim. A malicious application detection framework using automatic feature extraction tool on android market. In *3rd International Conference on Computer Science and Information Technology (ICCSIT'2013)*, pages 4–5, 2013.

[52] Jonathan Crussell, Clint Gibler, and Hao Chen. *Scalable semantics-based detection of similar Android applications*. ESORICS, 2013.

[53] Veelasha Moonsamy, Jia Rong, and Shaowu Liu. Mining permission patterns for contrasting clean and malicious android applications. *Future Generation Computer Systems*, 2013.

[54] You Joung Ham, Hyung-Woo Lee, Jae Deok Lim, and Jeong Nyeo Kim. Droidvulmonandroid based mobile device vulnerability analysis and monitoring system. In *Next Generation Mobile Apps, Services and Technologies (NGMAST), 2013 Seventh International Conference on*, pages 26–31. IEEE, 2013.

[55] Ilona Murynets and Roger Piqueras Jover. Anomaly detection in cellular machine-to-machine communications. In *Communications (ICC), 2013 IEEE International Conference on*, pages 2138–2143. IEEE, 2013.

[56] Zhemin Yang, Min Yang, Yuan Zhang, Guofei Gu, Peng Ning, and X Sean Wang. Appintent: Analyzing sensitive data transmission in android for privacy leakage detection. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*, pages 1043–1054. ACM, 2013.

[57] Wei Xu, Fangfang Zhang, and Sencun Zhu. Permlyzer: Analyzing permission usage in android applications. In *Software Reliability Engineering (ISSRE), 2013 IEEE 24th International Symposium on*, pages 400–410. IEEE, 2013.

[58] Steve Hanna, Ling Huang, Edward Wu, Saung Li, Charles Chen, and Dawn Song. Juxtapp: A scalable system for detecting code reuse among android applications. In *Detection of Intrusions and Malware, and Vulnerability Assessment*, pages 62–81. Springer, 2013.

[59] Dong-Jie Wu, Ching-Hao Mao, Te-En Wei, Hahn-Ming Lee, and Kuo-Ping Wu. Droidmat: Android malware detection through manifest and api calls tracing. In *Information Security (Asia JCIS), 2012 Seventh Asia Joint Conference on*, pages 62–69. IEEE, 2012.

[60] Hao Peng, Chris Gates, Bhaskar Sarma, Ninghui Li, Yuan Qi, Rahul Potharaju, Cristina Nita-Rotaru, and Ian Molloy. Using probabilistic generative models for ranking risks of android apps. In *Proceedings of the 2012 ACM conference on Computer and communications security*, pages 241–252. ACM, 2012.

[61] Yajin Zhou, Zhi Wang, Wu Zhou, and Xuxian Jiang. Hey, you, get off of my market: Detecting malicious apps in official and alternative android markets. In *Proceedings of the 19th Annual Network and Distributed System Security Symposium*, pages 5–8, 2012.

[62] Wu Zhou, Yajin Zhou, Xuxian Jiang, and Peng Ning. Detecting repackaged smartphone applications in third-party android marketplaces. In *Proceedings of the second ACM conference on Data and Application Security and Privacy*, pages 317–326. ACM, 2012.

[63] Asaf Shabtai, Uri Kanonov, Yuval Elovici, Chanan Glezer, and Yael Weiss. "andromaly": a behavioral malware detection framework for android devices. *Journal of Intelligent Information Systems*, 38(1): 161–190, 2012.

[64] Axelle Apvrille and Tim Strazzere. Reducing the window of opportunity for android malware gotta catch'em all. *Journal in Computer Virology*, 8(1–2):61–71, 2012.

[65] Abhijith Shastry, Murat Kantarcioglu, Yan Zhou, and Bhavani Thuraisingham. Randomizing smartphone malware profiles against statistical mining techniques. In *Data and Applications Security and Privacy XXVI*, pages 239–254. Springer, 2012.

[66] Te-En Wei, Ching-Hao Mao, Albert B Jeng, Hahn-Ming Lee, Horng-Tzer Wang, and Dong-Jie Wu. Android malware detection via a latent network behavior analysis. In *Trust, Security and Privacy in Computing and Communications (TrustCom), 2012 IEEE 11th International Conference on*, pages 1251–1258. IEEE, 2012.

[67] Borja Sanz, Igor Santos, Carlos Laorden, Xabier Ugarte-Pedrero, PabloGarcia Bringas, and Gonzalo Álvarez. Puma: Permission usage to detect malware in android. In Álvaro Herrero, Václav Snášel, Ajith Abraham, Ivan Zelinka, Bruno Baruque, Héctor Quintián, José Luis Calvo, Javier Sedano, and Emilio Corchado, editors, *International Joint Conference CISIS12-ICEUTE12-SOCO´12 Special Sessions*, volume 189 of *Advances in Intelligent Systems and Computing*, pages 289–298. Springer Berlin Heidelberg, 2013.

[68] Chao Yang, Vinod Yegneswaran, Phillip Porras, and Guofei Gu. Detecting money-stealing apps in alternative android markets. In *Proceedings of the 2012 ACM conference on Computer and communications security*, pages 1034–1036. ACM, 2012.

[69] Ingo Bente, Bastian Hellmann, Joerg Vieweg, Josef von Helden, and Gabi Dreo. Tcads: Trustworthy, context-related anomaly detection for smartphones. In *Network-Based Information Systems (NBiS), 2012 15th International Conference on*, pages 247–254. IEEE, 2012.

[70] Justin Sahs and Latifur Khan. A machine learning approach to android malware detection. In *Intelligence and Security Informatics Conference (EISIC), 2012 European*, pages 141–147. IEEE, 2012.

[71] Seung-Hyun Seo, Dong-Guen Lee, and Kangbin Yim. Analysis on maliciousness for mobile applications. In *Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2012 Sixth International Conference on*, pages 126–129. IEEE, 2012.

[72] Lena Chekina, Duku Mimran, Lior Rokach, Yuval Elovici, and Bracha Shapira. Detection of deviations in mobile applications network behavior. *arXiv preprint arXiv:1208.0564*, 2012.

[73] Muhamed Halilovic and Abdulhamit Subasi. Intrusion detection on smartphones. *arXiv preprint arXiv:1211.6610*, 2012.

[74] PENG Guojun, SHAO Yuru, WANG Taige, ZHAN Xian, and ZHANG Huanguo. Research on android malware detection and interception based on behavior monitoring. 17(5), 2012.

[75] Cong Zheng, Shixiong Zhu, Shuaifu Dai, Guofei Gu, Xiaorui Gong, Xinhui Han, and Wei Zou. Smartdroid: an automatic system for revealing ui-based trigger conditions in android applications. In *Proceedings of the second ACM workshop on Security and privacy in smartphones and mobile devices*, pages 93–104. ACM, 2012.

[76] Gianluca Dini, Fabio Martinelli, Ilaria Matteucci, Marinella Petrocchi, Andrea Saracino, and Daniele Sgandurra. A multi-criteria-based evaluation of android applications. In *Trusted Systems*, pages 67–82. Springer, 2012.

[77] Michael Grace, Yajin Zhou, Zhi Wang, and Xuxian Jiang. Systematic detection of capability leaks in stock android smartphones. In *Proceedings of the 19th Annual Symposium on Network and Distributed System Security*, 2012.

[78] Michael Grace, Yajin Zhou, Qiang Zhang, Shihong Zou, and Xuxian Jiang. Riskranker: scalable and accurate zero-day android malware detection. In *Proceedings of the 10th international conference on Mobile systems, applications, and services*, pages 281–294. ACM, 2012.

[79] Gianluca Dini, Fabio Martinelli, Andrea Saracino, and Daniele Sgandurra. Madam: a multi-level anomaly detector for android malware. In *Computer Network Security*, pages 240–253. Springer, 2012.

[80] Lingguang Lei, Yuewu Wang, Jiwu Jing, Zhongwen Zhang, and Xingjie Yu. Meaddroid: detecting monetary theft attacks in android by dvm monitoring. In *Information Security and Cryptology–ICISC 2012*, pages 78–91. Springer, 2013.

[81] Jordi Cucurull, Simin Nadjm-Tehrani, and Massimiliano Raciti. Modular anomaly detection for smartphone ad hoc communication. In *Information Security Technology for Applications*, pages 65–81. Springer, 2012.

[82] Kejun Xin, Gang Li, Zhongyuan Qin, and Qunfang Zhang. Malware detection in smartphone using hidden markov model. In *Multimedia Information Networking and Security (MINES), 2012 Fourth International Conference on*, pages 857–860. IEEE, 2012.

[83] Hua Zha and Chunlin Peng. Method of smartphone users' information protection based on composite behavior monitor. In *Intelligent Computing Technology*, pages 252–259. Springer, 2012.

[84] Chanmin Yoon, Dongwon Kim, Wonwoo Jung, Chulkoo Kang, and Hojung Cha. Appscope: Application energy metering framework for android smartphone using kernel activity monitoring. In *USENIX ATC*, 2012.

[85] You-Joung Ham, Won-Bin Choi, Hyung-Woo Lee, JaeDeok Lim, and Jeong Nyeo Kim. Vulnerability monitoring mechanism in android based smartphone with correlation analysis on event-driven activities. In *Computer Science and Network Technology (ICCSNT), 2012 2nd International Conference on*, pages 371–375. IEEE, 2012.

[86] Lok Kwong Yan and Heng Yin. Droidscope: seamlessly reconstructing the os and dalvik semantic views for dynamic android malware analysis. In *Proceedings of the 21st USENIX Security Symposium*, 2012.

[87] Iker Burguera, Urko Zurutuza, and Simin Nadjm-Tehrani. Crowdroid: behavior-based malware detection system for android. In *Proceedings of the 1st ACM workshop on Security and privacy in smartphones and mobile devices*, pages 15–26. ACM, 2011.

[88] Hahnsang Kim, Kang G Shin, and Padmanabhan Pillai. Modelz: monitoring, detection, and analysis of energy-greedy anomalies in mobile handsets. *Mobile Computing, IEEE Transactions on*, 10(7): 968–981, 2011.

[89] Hsiu-Sen Chiang and W Tsaur. Identifying smartphone malware using data mining technology. In *Computer Communications and Networks (ICCCN), 2011 Proceedings of 20th International Conference on*, pages 1–6. IEEE, 2011.

[90] Bryan Dixon, Yifei Jiang, Abhishek Jaiantilal, and Shivakant Mishra. Location based power analysis to detect malicious code in smartphones. In *Proceedings of the 1st ACM workshop on Security and privacy in smartphones and mobile devices*, pages 27–32. ACM, 2011.

[91] Zhang Lei, Zhu Junmao, Tian Zhongguang, Liu Yulong, and Wang Tao. Design of mobile phone security system based on detection of abnormal

behavior. In *Proceedings of the 2011 First International Conference on Instrumentation, Measurement, Computer, Communication and Control*, pages 479–482. IEEE Computer Society, 2011.

[92] Amir Houmansadr, Saman A Zonouz, and Robin Berthier. A cloud-based intrusion detection and response system for mobile phones. In *Dependable Systems and Networks Workshops (DSN-W), 2011 IEEE/IFIP 41st International Conference on*, pages 31–32. IEEE, 2011.

[93] Peter Gilbert, Byung-Gon Chun, Landon P Cox, and Jaeyeon Jung. Vision: automated security validation of mobile apps at app markets. In *Proceedings of the second international workshop on Mobile cloud computing and services*, pages 21–26. ACM, 2011.

[94] Erika Chin, Adrienne Porter Felt, Kate Greenwood, and David Wagner. Analyzing inter-application communication in android. In *Proceedings of the 9th international conference on Mobile systems, applications, and services*, pages 239–252. ACM, 2011.

[95] Leonid Batyuk, Markus Herpich, Seyit Ahmet Camtepe, Karsten Raddatz, A-D Schmidt, and Sahin Albayrak. Using static analysis for automatic assessment and mitigation of unwanted and malicious activities within android applications. In *Malicious and Unwanted Software (MALWARE), 2011 6th International Conference on*, pages 66–72. IEEE, 2011.

[96] Takamasa Isohara, Keisuke Takemori, and Ayumu Kubota. Kernel-based behavior analysis for android malware detection. In *Computational Intelligence and Security (CIS), 2011 Seventh International Conference on*, pages 1011–1015. IEEE, 2011.

[97] Lei Liu and Dai Ping Li. Analysis based on of android malicious code intrusion detection. *Advanced Materials Research*, 756: 3924–3928, 2013.

[98] Francesco Di Cerbo, Andrea Girardello, Florian Michahelles, and Svetlana Voronkova. Detection of malicious applications on android os. In *Computational Forensics*, pages 138–149. Springer, 2011.

[99] Liang Xie, Xinwen Zhang, Jean-Pierre Seifert, and Sencun Zhu. pbmds: a behavior-based malware detection system for cellphone devices. In *Proceedings of the third ACM conference on Wireless network security*, pages 37–48. ACM, 2010.

[100] Markus Jakobsson and Karl-Anders Johansson. Retroactive detection of malware with applications to mobile platforms. In *Proceedings of the 5th USENIX conference on Hot topics in security*, pages 1–13. USENIX Association, 2010.

[101] Asaf Shabtai, Uri Kanonov, and Yuval Elovici. Intrusion detection for mobile devices using the knowledge-based, temporal abstraction method. *Journal of Systems and Software*, 83(8): 1524–1537, 2010.

[102] Thomas Blasing, Leonid Batyuk, A-D Schmidt, Seyit Ahmet Camtepe, and Sahin Albayrak. An android application sandbox system for suspicious software detection. In *Malicious and Unwanted Software (MALWARE), 2010 5th International Conference on*, pages 55–62. IEEE, 2010.

[103] Asaf Shabtai and Yuval Elovici. Applying behavioral detection on android-based devices. In *Mobile Wireless Middleware, Operating Systems, and Applications*, pages 235–249. Springer, 2010.

[104] Asaf Shabtai, Yuval Fledel, and Yuval Elovici. Automated static code analysis for classifying android applications using machine learning. In *Computational Intelligence and Security (CIS), 2010 International Conference on*, pages 329–333. IEEE, 2010.

[105] William Enck, Peter Gilbert, Byung-Gon Chun, Landon P Cox, Jaeyeon Jung, Patrick McDaniel, and Anmol Sheth. Taintdroid: An information-flow tracking system for realtime privacy monitoring on smartphones. In *OSDI*, volume 10, pages 1–6, 2010.

[106] Georgios Portokalidis, Philip Homburg, Kostas Anagnostakis, and Herbert Bos. Paranoid android: versatile protection for smartphones. In *Proceedings of the 26th Annual Computer Security Applications Conference*, pages 347–356. ACM, 2010.

[107] Tansu Alpcan, Christian Bauckhage, and Aubrey-Derrick Schmidt. A probabilistic diffusion scheme for anomaly detection on smartphones. In *Information Security Theory and Practices. Security and Privacy of Pervasive Systems and Smart Devices*, pages 31–46. Springer, 2010.

[108] Fudong Li, Nathan Clarke, Maria Papadaki, and Paul Dowland. Behaviour profiling on mobile devices. In *Emerging Security Technologies (EST), 2010 International Conference on*, pages 77–82. IEEE, 2010.

[109] Ashkan Sharifi Shamili, Christian Bauckhage, and Tansu Alpcan. Malware detection on mobile devices using distributed machine learning. In *Pattern Recognition (ICPR), 2010 20th International Conference on*, pages 4348–4351. IEEE, 2010.

[110] William Enck, Machigar Ongtang, and Patrick McDaniel. On lightweight mobile phone application certification. In *Proceedings of the 16th ACM conference on Computer and communications security*, pages 235–245. ACM, 2009.

[111] A-D Schmidt, Rainer Bye, H-G Schmidt, Jan Clausen, Osman Kiraz, Kamer A Yuksel, Seyit Ahmet Camtepe, and Sahin Albayrak. Static analysis of executables for collaborative malware detection on android. In *Communications, 2009. ICC'09. IEEE International Conference on*, pages 1–5. IEEE, 2009.

[112] Lei Liu, Guanhua Yan, Xinwen Zhang, and Songqing Chen. Virusmeter: Preventing your cellphone from spies. In *Recent Advances in Intrusion Detection*, pages 244–264. Springer, 2009.

[113] Jong-seok Lee, Tae-Hyung Kim, and Jong Kim. Energy-efficient run-time detection of malware-infected executables and dynamic libraries on mobile devices. In *Future Dependable Distributed Systems, 2009 Software Technologies for*, pages 143–149. IEEE, 2009.

[114] Aubrey-Derrick Schmidt, Frank Peters, Florian Lamour, Christian Scheel, Seyit Ahmet Çamtepe, and Şahin Albayrak. Monitoring smartphones for anomaly detection. *Mobile Networks and Applications*, 14(1): 92–106, 2009.

[115] Liang Xie, Xinwen Zhang, Ashwin Chaugule, Trent Jaeger, and Sencun Zhu. Designing system-level defenses against cellphone malware. In *Reliable Distributed Systems, 2009. SRDS'09. 28th IEEE International Symposium on*, pages 83–90. IEEE, 2009.

[116] Aubrey-Derrick Schmidt, Hans-Gunther Schmidt, Jan Clausen, Kamer A Yuksel, Osman Kiraz, Ahmet Camtepe, and Sahin Albayrak. Enhancing security of linux-based android devices. In *in Proceedings of 15th International Linux Kongress. Lehmann*, 2008.

[117] Abhijit Bose, Xin Hu, Kang G Shin, and Taejoon Park. Behavioral detection of malware on mobile handsets. In *Proceedings of the 6th international conference on Mobile systems, applications, and services*, pages 225–238. ACM, 2008.

[118] Hahnsang Kim, Joshua Smith, and Kang G Shin. Detecting energy-greedy anomalies and mobile malware variants. In *Proceedings of the 6th international conference on Mobile systems, applications, and services*, pages 239–252. ACM, 2008.

[119] Deepak Venugopal and Guoning Hu. Efficient signature based malware detection on mobile devices. *Mobile Information Systems*, 4(1): 33–49, 2008.

[120] Timothy K Buennemeyer, Theresa M Nelson, Lee M Clagett, John Paul Dunning, Randy C Marchany, and Joseph G Tront. Mobile device profiling and intrusion detection using smart batteries. In *Hawaii International Conference on System Sciences, Proceedings of the 41st Annual*, pages 296–296. IEEE, 2008.

Biographies

Abdullah Alzahrani is a PhD candidate at the faculty of Computer Science, University of New Brunswick, Canada. He is a lecturer at Computer Science and computer Engineering department, University of Hail, Saudi Arabia. His research interests include botnet detection, Android security, network security, malware analysis and reverse engineering. He is a member of the Information Security Centre of Excellence, University of New Brunswick.

Dr. Natalia Stakhanova is the New Brunswick Innovation Research Chair in Cyber Security at University of New Brunswick, Canada. Her research interests include intrusion detection and response, smartphone security, security assessment and generally network and computer security. Natalia Stakhanova was the recipient of the Nokia Best Student Paper Award at The IEEE

International Conference on Advanced Information Networking and Applications (AINA). She served on the program committee of several conferences and workshops in area of information security and assurance, including the Conference on Privacy, Security and Trust (PST). Natalia developed a number of technologies that have been adopted by high-tech companies such as IBM and she currently has three patents in the field of computer security.

Hugo Gonzalez is a PhD student at the Information Security Centre of Excellence, University of New Brunswick, Canada. He is a faculty member of the Polytechnic University of San Luis Potosi, Mexico. His current research interests include network security and malware analysis. He is a member of the Association for Computing Machinery, the IEEE Computer Society and The Honeynet Project.

Ali Ghorbani has held a variety of positions in academia for the past 34 years. He currently serves as Dean of the Faculty of Computer Science and Founding Director of the Information Security Centre of Excellence at the

University of New Brunswick (UNB), Fredericton, Canada. Dr. Ghorbani is the co-Editor-In-Chief of Computational Intelligence, an international journal. He supervised more than 150 research associates, postdoctoral fellows, and undergraduate & graduate students and authored more than 250 research papers in journals and conference proceedings and has edited 11 volumes. He is the co-inventor of 3 patents in the area of Network Security and Web Intelligence. In 2012 he spawn off "Ara Labs Security Solutions" and "Eyesover Technologies". His current research focus is Network & Information Security, Complex Adaptive Systems, Critical Infrastructure Protection, and Web Intelligence. His book, Intrusion Detection and Prevention Systems: Concepts and Techniques, published by Springer in October 2009. Dr. Ghorbani is the Senior member of IEEE and the member of ACM, and Canadian Information Processing Society (CIPS). He is also the coordinator of the Privacy, Security and Trust (PST*net) research network.